

(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(51) . Int. Cl.⁸
H04N 7/24 (2006.01)

(45) 공고일자 2006년02월28일
(11) 등록번호 10-0536980
(24) 등록일자 2005년12월08일

(21) 출원번호	10-1998-0012598	(65) 공개번호	10-1998-0081237
(22) 출원일자	1998년04월09일	(43) 공개일자	1998년11월25일

(30) 우선권주장 08/833,776 1997년04월09일 미국(US)

(73) 특허권자 휴렛-팩커드 컴퍼니(엘라웨어주법인)
미합중국 캘리포니아주 (우편번호 94304) 팔로 알토 하노버 스트리트 3000

(72) 발명자 러스트 로버트 에이
미국 아이다호주 83704 보이세 알라마 5061

(74) 대리인 김창세
장성구

심사관 : 김홍수

(54) 데이터 압축 방법, 데이터 압축해제 방법 및 데이터 압축 장치

요약

본 발명은 압축 실행의 개시 시에 통계치(statistics)를 신속히 추적하여, 데이터의 크기가 증가함에 따라 기준의 페이스(pace)로 늦춤(slowing down)으로써, 보다 큰 데이터 세트의 압축율에 영향을 작게 하면서 작은 데이터 세트에 대해 개선된 압축율을 제공한다. 이것은 개시 시에 확률표(probability table)의 크기를 제한함으로써 수행된다. 압축기를 통해 보다 많은 데이터가 통과할수록 표의 크기는 확장된다. 확률의 크기는 데이터 세트의 개시 시에 제한된 수의 바이트에 대한 콘텍스트 윈도우(context window)를 점차 개방함으로써 제어된다.

대표도

도 1

명세서

도면의 간단한 설명

도 1은 산술 압축기의 블록도,

도 2는 사용된 확률표의 크기 증가를 그래픽으로 도시한 도면,

도 3은 주어진 콘텍스트 모델을 이용하여, 사용된 확률표의 크기를 동적으로 조정하는 방법을 도시한 도면,

도 4는 바람직한 실시예의 논리 동작의 순서도,

도 5는 본 발명에 따른 산술 압축기의 블록도.

도면의 주요 부분에 대한 부호의 설명

101 : 비트 1101 : 이전 행의 이미지

1102 : 카운터 1107 : 이미지/시프트 레지스터

1113 : 확률표 1114 : 압축기

1115 : 콘텍스트 모델 1116 : 산술 압축기

발명의 상세한 설명

발명의 목적

발명이 속하는 기술 및 그 분야의 종래기술

본 발명은 산술 부호화(arithmetic encoding) 방법 및 장치에 관한 것으로서, 보다 구체적으로는, 보다 많은 양의 데이터에 대해서는 높은 압축율을 유지하면서 적은 양의 데이터에 대한 적응을 가속화하는 콘텍스트 모델(context model)에 관한 것이다.

디지털 데이터 신호의 스트림을 압축된 디지털 코드 신호로 부호화하고, 압축된 디지털 코드 신호를 원래 데이터로 다시 복호화하는 데이터 압축 시스템은 종래의 기술로서 알려져 있다. 데이터 압축은 주어진 포맷의 데이터를 원래의 공간보다 적은 공간을 필요로 하는 대안적인 포맷으로 변환하고자 하는 임의의 처리를 지칭한다. 데이터 압축 시스템의 목적은 주어진 형태의 디지털 정보를 유지하는 데 필요한 저장 공간이나, 그것을 전송하는 데 필요한 시간을 효율적으로 절약하는 것이다.

실제 이용을 위해서는, 범용 디지털 데이터 압축 시스템은 소정의 기준을 만족시켜야 한다. 이 데이터 압축 시스템은 교환성(reciprocity)을 가져야 한다. 데이터 압축 시스템이 교환성 특성을 갖기 위해서는, 정보의 변경이나 손실없이 압축 데이터를 자신의 원래 형태로 재확장(re-expand) 또는 복호화할 수 있어야 한다. 복호화된 데이터와 원래 데이터는 동일해야 하며, 서로를 구별할 수 없어야 한다. 교환성 특성은 정보 이론(information theory)에서 사용되는 엄격한 무잡음성과 동일한 의미이다. 몇몇 응용에서는 교환성 특성을 엄격하게 고수할 필요가 없다. 이것은 특히 그래픽 데이터를 다루는 경우의 응용에서 그러하다. 인간의 시각은 잡음에 민감하지 않기 때문에, 압축/압축해제 처리 동안 발생하는 약간의 정보 변경 또는 손실은 용인될 수 있다.

이 시스템은 데이터 압축 및 압축해제 시스템이 통신하고 있는 장치에 의해 제공되고 용인되는 데이터 속도에 대하여 충분한 성능을 제공해야 한다. 데이터가 압축될 수 있는 비율은 압축 시스템의 입력 데이터의 처리율에 의해 결정되며, 전형적으로는 초당 수백만 바이트(megabytes/sec)이다. 전형적으로 데이터 속도가 1 megabyte/sec를 초과하는 오늘날의 디스크, 테이프 및 통신 시스템에서 달성되는 데이터 속도를 유지하기 위해서는 충분한 성능을 갖고 있어야 한다. 따라서, 데이터 압축 및 압축해제 시스템은 전체 시스템에 나쁜 영향을 미치지 않도록 충분한 데이터 대역폭을 가져야 한다. 전형적으로, 데이터 압축 및 압축해제 시스템의 성능은 압축 및 압축해제에 필요한 연산과, 통계 데이터를 저장하고 압축 및 압축해제를 관리하는 데 이용되는 RAM(random access memory) 등과 같은 시스템 구성소자의 속도에 의해 제한된다. 압축 장치의 성능은 압축기의 입력 문자당 필요한 프로세서 주기(processor cycles)의 수에 의해 특징지어진다. 주기의 수가 작을수록 성능이 개선된다.

데이터 압축 및 압축해제 시스템 설계 시의 다른 중요한 기준은 압축율(compression ratio)에 의해 특징지어지는 압축 효율이다. 압축율은 압축되지 않은 형태의 데이터 크기를 압축된 형태의 데이터 크기로 나눈 비율이다. 데이터가 압축되려면, 데이터는 리던던시를 포함해야 한다. 압축 효율은 압축 처리 시 얼마나 효율적으로 입력 데이터의 리던던시를 이용하

는지에 따라 결정된다. 전형적인 컴퓨터 저장 데이터에서, 리던던시는 각각의 심볼, 예를 들어, 디지트(digits), 바이트(bytes) 또는 문자(characters)를 불규칙하게 사용하는 경우와, 공통 워드(words), 여백 레코드 필드 등과 같은 심볼 시퀀스가 빈번하게 재현되는 경우 모두에서 발생한다.

종래 기술의 범용 데이터 압축 절차도 또한 종래기술 분야에서 잘 알려져 있으며, 세 가지 관련 절차로 허프만 방법(Huffman method), 턴스탈 방법(Tunstall method) 및 렘펠-지브 방법(Lempel-Ziv method)이 있다. 허프만 방법은 널리 알려져 사용되고 있으며, 이에 관한 내용은 D. A. Huffman에 의해 기술된 "A Method For Construction Of Minimum Redundancy Codes"란 명칭의 Proceedings IRE, 40, 10 pages 1098-1100(Sept. 1952)의 문헌에 기술된 논문을 참조하면 알 수 있다. 턴스탈 알고리즘은 B. P. Tunstall에 의해 기술된 "Synthesis of Noiseless Compression Codes"란 명칭의 Georgia Institute of Technology(Sept. 1967)의 문헌에 기술된 박사 학위 논문을 참조하면 알 수 있다. 렘펠-지브 절차는 J. Ziv 및 A. Lempel에 의해 기술된 "A Universal Algorithm For Sequential Data Compression"이라는 명칭의 IEEE Transactions on Information Theory, IT-23, 3, pages 337-343(May, 1977)의 문헌을 참조하면 알 수 있다.

최초의 범용 데이터 압축 절차들 중 하나는 허프만 방법이다. 간단히 설명하면, 허프만 절차는 전체 길이의 심볼 세그먼트를 가변 길이 워드(variable length words)로 매핑(mapping)한다. 허프만 데이터 압축 절차는 두 가지 제한을 갖고 있다. 첫째, 허프만 절차는 압축될 입력 데이터가 고정 길이의 심볼 세그먼트로 분석(parsing)된다는 제약 조건 하에서 동작한다. 허프만 절차는 이 제약 조건 하에서 얻을 수 있는 최적의 압축율을 제공하지만, 제약 조건이 완화될 경우에는 다른 절차를 이용하면 더욱 현저히 높은 압축율을 얻을 수 있다. 둘째, 허프만 코딩은 소스 데이터의 통계적 특성을 모두 알고 있어야 한다. 허프만 절차는 각각의 고정 길이 입력 세그먼트가 나타날 확률을 알고 있다는 가정 하에서 동작한다. 실제로, 이와 같은 허프만 절차의 요건은 데이터 처리 동안 필요한 통계치를 측정하는 적응적인 버전(version)의 절차를 이용함으로써 충족될 수 있다. 그러나, 이러한 방법은 번거로운 일이고, 상당한 작업 메모리 공간을 필요로 하며, 적응 동안에 차선으로(sub-optimally) 수행된다.

턴스탈 알고리즘은 가변 길이의 심볼 세그먼트를 고정 길이의 이진 워드에 매핑하기 때문에, 고정 길이의 제약 조건이 입력 세그먼트 대신 출력 세그먼트에 적용되며, 허프만 절차를 보완한 것이다. 허프만 절차와 마찬가지로, 턴스탈 절차는 소스 데이터의 확률에 대한 사전 지식이 필요하다. 이 경우에도, 이러한 사전 지식에 대한 요건은 데이터를 처리하는 동안 통계치를 측정하는 적응적인 버전을 이용함으로써 어느 정도 충족될 수 있다.

렘펠-지브 절차는 가변 길이의 심볼 세그먼트를 가변 길이의 이진 워드에 매핑한다. 이 절차는 입력 또는 출력 세그먼트에 대한 제약 조건이 없을 때 점근적으로(asymptotically) 최적이다. 이 절차에서는 입력 데이터 스트링(data string)이 적응적으로 성장된 세그먼트로 분석되며, 각각의 세그먼트는 입력 데이터로부터의 하나의 새로운 심볼을 접미시킨 입력 스트링의 조기 부분의 정확한 복제본(copy)으로 구성된다. 생성될 복제본은 가능한 한 가장 길며, 조기에 분석된 임의의 세그먼트와 일치하도록 제한되지는 않는다. 출력의 세그먼트로 대체하는 코드 워드는 조기에 복사된 부분의 개시 포인터, 복제본의 길이 및 새로운 심볼로 구성된 정보를 포함한다.

허프만 또는 쇼논-페노(Shannon-Fano) 코딩은 완벽한 데이터 압축 수단처럼 보일 수 있다. 그러나, 실제는 그렇지 않다. 전술한 바와 같이, 이들 코딩 방법은 심볼의 확률이 $1/2$ 의 정수 거듭제곱(integral power)일 경우에만 최적인 방법으로, 통상적으로는 완벽한 수단이 아니다.

산술 코딩 기법은 이러한 제한 조건을 갖지 않는다. 즉, 이 기법은 메시지를 하나의 단일 유닛으로서 메시지를 처리하는 것과 동일한 효과를 달성하며(허프만 코딩의 경우, 단일의 가능한 모든 메시지를 열거해야 함), 이에 따라, 임의의 소스에 대한 압축 효과에 대해 이론적인 엔트로피 경계값(theoretical entropy bound)을 얻을 수 있다.

산술 코딩에서는, 하나의 결정이 다른 결정 이후에 부호화되어, 번호 라인을 따라 보다 작고 적은 포함 간격이 성공적으로 정의된다. 산술 코딩에 대한 추가적인 정보는 G. G. Langdon, Jr.에 의해 기술된 "An Introduction To Arithmetic Encoding"이란 명칭의 IBM Journal of Research and Development, Vol. 28, n. 2, March 1984, 135-149의 문헌과, D. R. Helman, G. G. Langdon, Jr. 및 J. J. Rissanen에 의해 기술된 "Arithmetic Compression Code Control Parameters Approximation"이란 명칭의 Vol. 23, n. 11, April 1981, 5112-5114의 문헌과, Langdon, Jr. 등에 의해 기술된 "Arithmetic Coding Encoder And Decoder System"이란 명칭의 미국 특허 제 4,905,297 호에 개시되어 있으며, 이들 모두는 본 명세서에서 참조로 인용된다.

위의 문헌에 기술된 바와 같이, 산술 코딩에서 각각의 결정은 다수의 가능한 베타적 결과 즉 "이벤트(events)"를 갖는다. 각각의 결과 즉 이벤트는 데이터에서 심볼로 표현된다. 이미징 환경에서, 예를 들어, 각각의 결정은 주어진 픽셀(pixel)이

검정색인지 아닌지의 여부에 대응할 수 있다. 결정 결과는 픽셀이 검정색인 경우에는 Y(또는 YES) 심볼로 표현되고, 픽셀이 검정색이 아닌 경우에는 N(또는 NO) 심볼로 표현된다. 따라서, 다수의 결정은 심볼 시퀀스, 예를 들면, YNNY...로 표현될 수 있다.

종래의 산술 코딩 기법에 따르면, 확률 라인 상에는 현재의 간격이 정의된다. 제 1 현재 간격은 0 내지 1이다. 현재의 간격은 다음 결정에 대해 가능한 하나의 결과에 대응하는 세그먼트로 분할된다. 각각의 결정에 대해서 단지 두 개의 가능한 결과만이 존재하는 경우, 현재의 간격은 두 개의 세그먼트로 분할된다. 각각의 세그먼트의 길이는 세그먼트 자신과 각각 관련된 확률을 기초로 한다. 각각의 확률은 고정된 상태로 유지하거나, 혹은 결정 데이터가 입력될 때 적응될 수 있다.

보다 높은 빈도로 발생하는 심볼에 대한 큰 세그먼트의 상관은 압축 효과를 발생시킨다. 앞서 인용된 문헌("An Introduction To Arithmetic Encoding")에는 4개의 심볼에 대한 산술 코딩의 예가 개시되어 있으며, 이 문헌에서 각각의 결정은 "a" 이벤트(50%의 확률), "b" 이벤트(25%의 확률), "c" 이벤트(12.5%의 확률) 또는 "d" 이벤트(12.5%의 확률)를 생성할 수 있다. 4개의 이벤트를 이진 형태로 표현하려면, 각각의 결정에 대해 2 비트가 필요하며, 각각의 이벤트는 00, 01, 10, 11로 표현된다. "aab"와 같은 3 개의 결정에서, 순방향 부호화되는 데이터는 6 비트인 00 00 01일 수 있다. 그러나, 이 문헌의 137 페이지에 기술된 바와 같이, 산술 코딩 방안은 시퀀스 "aab"를 값 0.001로 표현할 수 있다. 정보를 6 비트가 아닌 3 비트로 표현할 수 있다. 이러한 비트의 보존(conservation)은 관련된 확률이 비교적 높은 연속적인 이벤트가 발생될 때 이루어진다.

발생 확률이 낮고 비교적 짧은 라인 세그먼트를 갖는 이벤트가 많이 발생하는 경우에는 보존이 나빠진다. 전술한 확률에 의하면, 이벤트 시퀀스 "dd"는 11 11과 같은 부호화된 데이터로 표현되나, 산술 코딩에 의해, "dd" 이벤트는 111111로 표현된다. 대응하여 보다 높은 빈도로 발생하는 이벤트에 실질적으로 대응하는 보다 큰 세그먼트가 제공된다면, 발생 확률이 높은 심볼이 생성될 때 보존이 이루어짐에 따라 발생 확률이 낮은 심볼에 필요한 추가적인 비트도 중요하다.

산술 코딩은 전체 압축 실행을 통한 데이터에 대해 적응적이며, 예전의 데이터를 결코 잊지 않는다. 이것은 일정하게 사전(dictionary)의 내용을 상실하는 다수의 LZ 기반 방안들과는 다르다. LZ는 사전을 재구축하여, 다음 데이터 셋션에 적응하도록 한다. LZ 방안은 100KB의 데이터뿐만 아니라 1KB의 데이터도 압축한다. 산술 코딩은 실행을 통해 그 확률을 계속해서 개선시키며, 이는 산술 코딩이 자신의 적응을 최적화할 기회를 갖지 못하기 때문에 1KB의 데이터 상에서 마찬가지로 수행되지 않음을 의미한다. 그러나, 산술 코딩은 1KB를 LZ보다는 더 양호하게 압축한다.

산술 코딩은 확률표(probability table)를 이용하여 이미지에 대한 통계치를 저장한다. 각각의 비트가 압축됨에 따라, 비트에 대한 처리 방법을 결정하기 위해 표에 액세스한다. 표가 클수록, 최종/최적의 상태로 더욱 늦게 이동한다. 그러나, 표가 클수록, 각각의 비트에 대해 더 많은 정보를 이용할 수 있기 때문에 더 큰 이미지에 대해 압축율은 더 좋아진다. 간단한 실험에 의하면, 더 큰 표로부터의 이점에 대한 분기점(break point)은 대략 10KB이며, 이러한 분기점 이후, 더 큰 표는 압축율을 크게 개선한다.

프린터 동작 시, 몇 가지 상이한 유형의 이미지가 생성된다. 어떤 이미지는 크기가 100KB 내지 200KB이며, 어떤 이미지는 400B 또는 그 이하의 작은 크기를 갖기도 한다. 이를 작은 이미지는 주로 폰트 캐시(font cache)에 의한 것이다. 각각의 고유한 문자는 처음에 생성되어 프린트가 시작되기 전에 폰트 캐시에 저장된다. 페이지에 따라, 폰트 캐시에 의해 사용되는 공간의 양은 페이지의 성공적인 표현을 위한 사안이다.

LAN 환경에서 동일한 프린터가 사용되는 경우에는 폰트의 사용은 더욱 중요해진다. 몇몇 다른 사용자들은 그들이 자신이 선호하는 폰트 및 포인트 크기로 프린트 작업을 전송할 수 있다. 각각의 새로운 작업 시, 프린터는 요청된 문자 세트가 이전 작업으로부터의 폰트 캐시에 이미 존재하는지를 결정한다. 그것이 존재하지 않으면, 문자를 표현하는 데에는 시간이 걸린다. 프린터의 메모리가 폰트 캐시 문자로 채워짐에 따라, 그러한 문자가 요구에 즉각적으로 응하지 못하는 시점은 다른 문자를 위한 공간을 위해 폰트 캐시로부터 제거되어야 한다. 따라서, 더 긴 폰트 캐시 문자가 머물수록 - 특히, 몇몇 작업을 통해 - 더 빠른 사용자가 그들의 프린트 출력을 받게 된다. 폰트 캐시를 생성하기 위해 상당히 오랜 시간을 기다리는 것은 일반적이지 않다. 이것은 전원을 켜 후에 처음 페이지를 프린트할 때 흔히 알 수 있다.

발명이 이루고자 하는 기술적 과제

본 발명을 수행하기 위해서 데이터를 압축 및 압축해제하는 방법이 제공된다. 본 발명의 방법은, 처리된 데이터의 양의 카운트에 기초하여 콘텍스트 모델을 선택함으로써 수행된다. 산술 압축기는 콘텍스트 모델을 이용하여 데이터의 일부를 압

축하거나 압축해제한다. 카운터는 처리된 데이터의 양을 카운트한다. 압축 단계 후에는 카운터가 증가된다. 카운트가 사전 결정된 양보다 적으면 제 1 콘텍스트 모델이 선택되고, 카운트가 사전결정된 양보다 많으면 제 2 콘텍스트 모델이 선택된다. 임의의 수의 콘텍스트 모델을 정의할 수 있으며, 적절한 결정 포인트를 선택할 수 있다.

또한, 데이터를 압축 또는 압축해제하기 위한 장치가 제공되며, 이 장치는 확률표, 제 1 콘텍스트 모델 및 제 2 콘텍스트 모델을 포함하는 산술 압축기로 형성된다. 산술 압축기에 접속된 시프트 레지스터가 제공된다. 시프트 레지스터는 데이터를 수신한다. 산술 압축기에는 카운터가 접속된다. 데이터의 일부분이 사전결정된 양보다 작은 경우 카운터는 제 1 콘텍스트 모델을 이용하여 데이터의 일부분을 압축하도록 산술 압축기에 신호를 제공하고, 대안으로는, 제 2 콘텍스트 모델을 이용하여 데이터의 일부분을 압축하도록 산술 압축기에 신호를 제공한다.

발명의 구성 및 작용

이하의 상세한 설명을 첨부된 도면과 함께 고려함으로써 본 발명을 보다 잘 이해할 수 있을 것이다.

본 발명은 본 명세서에서 기술된 특정한 실시예로 국한되지 않는다. 도 1을 참조하면, 본 발명의 바람직한 실시예의 하드웨어 구현에의 블록도가 도시되어 있다. 산술 압축기(1116)는 비트(101)를 압축하고자 하며, 이미지/시프트 레지스터(1107)로부터의 데이터는 콘텍스트 모델(1115)로 전달된다. 콘텍스트 모델(1115)은 이미지/시프트 레지스터(1107)로부터의 데이터를 확률표(1113)에 매핑한다. 압축기는 확률표(1113) 및 콘텍스트 모델(1115)과 함께 비트(101)를 압축한다. 그 후, 압축 데이터는 일반적으로 메모리 장치(도시되지 않음)에 기록된다. 비트(101)는 이미지/시프트 레지스터(1107)로 시프트되며, 이미지로부터의 새로운 비트가 (101)로 시프트된다.

일반적으로, 압축해제는 압축과 동일한 방식으로 수행된다. 그러나, 압축해제 동안, 산술 압축기는 확률표(1113) 및 콘텍스트 모델(1115)을 이용하여 압축 데이터를 판독하고 압축해제한 후 비트(101)를 저장한다. 전술한 바와 같이, 일단 비트(101)가 압축해제되면, 이미지/시프트 레지스터(1107)의 데이터는 왼쪽으로 시프트된다. 이미지/시프트 레지스터(1107)로부터 출력된 데이터는 일반적으로 메모리 장치에 저장된다.

압축기(1114)는 이미지를 압축하거나 압축해제할 때 확률표(1113)를 이용하여 이미지에 대한 통계치를 저장한다. 각각의 비트가 압축/압축해제될 때, 이 비트를 처리하는 방법을 결정하기 위해서 확률표(1113)를 액세스한다. 확률표(1113)가 클수록, 최종/최적의 상태에 도달하는 시간이 더 오래 걸린다. 그러나, 확률표(1113)가 클수록 비트(101)에 대한 보다 많은 정보를 이용할 수 있으므로, 더 큰 이미지에 대해 압축율이 개선될 것이다.

콘텍스트 모델(1115)은 압축된 비트 주변의 데이터의 비트를 관찰함으로써 인덱스(index)를 확률표(1113)에 생성한다. 인덱싱에 대한 중요한 측면은 어드레싱된 위치가 부호화/복호화되는 비트의 값에 관한 유용한 정보를 포함하고 있다는 것이다. 구체적으로는, 확률표를 인덱싱하는 데 사용되는 비트는 부호화/복호화될 수 있는 비트를 신뢰성 있게 예측할 수 있도록 중요한 정보를 제공해야 한다. 예측에 대한 신뢰성이 높을수록, 이미지 압축 능력이 개선된다.

본 발명은, 압축 실행의 개시 시에 통계치를 신속하게 추적하여, 이미지의 크기가 증가함에 따라 기준의 페이스(pace)로까지 늦추어진다. 이것은 각각의 이미지의 개시 시에 확률표의 크기를 제한함으로써 달성된다. 압축기를 통해 보다 많은 데이터가 통과함에 따라 표의 크기는 확장된다. 확률의 크기는 처음의 5.1 킬로바이트의 이미지에 대한 콘텍스트 윈도우(context window)를 점차 개방함으로써 제어된다.

도 3을 참조하면, 더 많은 데이터가 압축됨에 따라 2 차원 콘텍스트 모델(300)이 확장된다. 각각의 퍽셀은 확률표의 어드레스 라인에 접속되어 있다. 퍽셀 상의 번호는 어떤 특정 어드레스 비트가 그 퍽셀에 의해 제어되고 있는지를 나타낸다. 도 3의 a를 참조하면, 처음의 100 바이트의 데이터에 대해, 어드레스 비트 9, 8, 7, 6, 1 및 0은 강제적으로 0으로 되며, 이것은 도 2의 a에 도시한 바와 같이 확률표의 크기를 효율적으로 감소시킨다. 다음의 1 킬로바이트 데이터는, 도 3의 b에도 시한 바와 같이 어드레스 비트 9, 8 및 0이 강제적으로 0으로 되어, 도 2의 b에 도시한 바와 같이 확률표의 크기를 효율적으로 감소시킨다. 다음의 4 킬로바이트 데이터의 경우, 도 3의 c에 도시한 바와 같이 어드레스 비트 0이 강제적으로 0으로 되어, 도 2의 c에 도시한 바와 같이 확률표의 크기를 효율적으로 감소시킨다. 그 후, 도 3의 d의 콘텍스트 모델과, 도 2의 d에 도시한 바와 같은 전체 확률표를 이용하여, 나머지 데이터를 압축한다.

도 3의 a의 초기의 콘텍스트 모델에서는 두 가지 측면이 중요하다. 첫째, 비트(301) 주변의 퍽셀들은 더 떨어져 있는 퍽셀보다 우위로 이용되어야 한다. 따라서, 어드레스 비트 5, 4, 3, 및 2가 이용된다. 둘째, 거의 모든 이미지에서, 1보다는 0이

더 흔하다. 그러므로, 마스크된 비트가 강제적으로 0이 되도록 하면, 이러한 양상은 마스크가 제거된 후의 압축율에도 도움이 된다. 다음 블록(도 3의 b)에서 비트(301)에 근접한 픽셀들은 마스크되어 있지 않다. 어드레스 비트 중 어느 것도 마스크되지 않는 최종 상태(도 3의 d)가 될 때까지 이러한 처리가 계속된다.

도 1의 블록도는 카운터를 추가하여 변형함으로써 본 발명을 수행할 수 있다. 도 5에 도시한 바와 같이, 카운터(1102)는 처리된 비트의 수를 카운트한다. 카운터(1102)는 어떤 비트를 마스크하는지를 나타내는 신호를 콘텍스트 모델(1115)에 출력한다. 추가적인 비트가 처리되면, 콘텍스트 모델(1115)은 전술한 바와 같이 더 적은 비트를 마스크하도록 지시를 받는다.

도 4에는 바람직한 실시예에 대한 흐름도가 도시되어 있다. 각각의 새로운 파일 또는 이미지가 처리되면, 카운터는 초기에 리셋된다(401). 압축될 비트가 데이터로부터 검색된다(403). 카운트가 X보다 작으면(405), 비트 압축 시(419) 소형 콘텍스트 모델이 이용된다(407). 유사한 방법으로, 카운트가 X보다 크고, Y보다 작으면(409) 비트 압축 시(419) 중형 콘텍스트 모델이 이용되고(411), 카운트가 Y보다 크고, Z보다 작으면(413) 비트 압축 시(419) 대형 콘텍스트 모델이 이용된다(415). 일단, Z보다 큰 바이트가 처리되면, 비트 압축 시(419) 전체 콘텍스트 모델이 이용된다(417). 압축 데이터는 저장된다(421). 카운트는 증가되며(423), 모든 데이터가 처리되었는지를 결정하는 체크가 행해진다. 다음 비트를 처리하기 전에, 시프트 레지스터내의 데이터는 시프트되어야 한다(426). 당업자라면 본 발명의 동작 순서를 변경하더라도, 기본적인 사상을 마찬가지로 구현할 수 있음을 이해할 것이다. 또한, 위에서는 소형, 중형, 대형 및 전체 콘텍스트 모델에 대해 기술하였지만, 소정의 수의 콘텍스트 모델을 정의할 수 있으며, X, Y, Z, ...에 대한 적절한 값을 선택할 수 있다.

본 발명은 보다 큰 이미지의 압축율에 영향을 작게 하면서, 폰트 캐시 데이터(즉, 작은 파일들)에 대하여 개선된 압축율을 제공한다. 라틴(Latin) 문자에 대한 특정한 경우, 폰트 캐시 문자의 압축은 40% 개선된다. 즉, 종래의 산술 코딩 압축 기법에 비해 문자가 40% 더 압축되어 폰트 캐시에 제공된다. 어드레스 비트를 마스킹하는 복잡도는 매우 낮으며, 설계시 약간의 논리가 부가될 뿐이다.

비록, 본 발명의 바람직한 실시예가 예시되고, 그러한 형태가 기술되었지만, 당업자라면 본 발명의 사상을 벗어나거나, 첨부된 특허 청구 범위의 영역을 벗어나지 않고서도 여러 가지 변형이 가능함을 알 수 있을 것이다.

발명의 효과

본 발명에 따른 데이터 압축/압축해제 방법 및 압축 장치에 따르면, 압축 실행의 개시 시에 통계치를 신속히 추적하여, 데이터의 크기가 증가함에 따라 기존의 페이스로 늦춤으로써, 보다 큰 데이터 세트의 압축율에 영향을 작게 하면서 작은 데이터 세트에 대해 개선된 압축율을 제공한다.

(57) 청구의 범위

청구항 1.

데이터를 압축하는 방법에 있어서,

처리된 상기 데이터의 양의 카운트에 기초하여 콘텍스트 모델(context model)을 선택하는 단계와,

상기 콘텍스트 모델을 이용하여 상기 데이터의 일부분을 압축하는 단계

를 포함하는 데이터 압축 방법.

청구항 2.

제 1 항에 있어서,

카운터를 이용하여 상기 처리된 데이터의 상기 양을 카운트하는 단계와,

상기 압축 단계 이후에 상기 카운터를 증가시키는 단계

를 더 포함하는 데이터 압축 방법.

청구항 3.

제 2 항에 있어서,

상기 콘텍스트 모델 선택 단계는, 상기 카운트가 사전결정된 양보다 적으면 제 1 콘텍스트 모델을 선택하고, 상기 양이 상기 사전결정된 양보다 적지 않으면 제 2 콘텍스트 모델을 선택하는 데이터 압축 방법.

청구항 4.

제 2 항에 있어서,

상기 데이터의 상기 일부분은 비트이고,

상기 처리된 데이터의 양은 1비트인 데이터 압축 방법.

청구항 5.

제 1 항에 있어서,

상기 데이터의 상기 압축된 일부분을 저장하는 단계를 더 포함하는 데이터 압축 방법.

청구항 6.

압축 데이터를 압축해제하는 방법에 있어서,

처리된 상기 압축 데이터의 양의 카운트에 기초하여 콘텍스트 모델을 선택하는 단계와,

상기 콘텍스트 모델을 이용하여 상기 압축 데이터의 일부분을 압축해제하는 단계

를 포함하는 데이터 압축해제 방법.

청구항 7.

제 6 항에 있어서,

카운터를 이용하여 상기 처리된 압축 데이터의 상기 양을 카운트하는 단계와,

상기 압축해제 단계 이후에 상기 카운터를 증가시키는 단계

를 더 포함하는 데이터 압축해제 방법.

청구항 8.

제 7 항에 있어서,

상기 콘텍스트 모델 선택 단계는 상기 카운트가 사전결정된 양보다 적으면 제 1 콘텍스트 모델을 선택하고, 상기 양이 상기 사전결정된 양보다 적지 않으면 제 2 콘텍스트 모델을 선택하는 데이터 압축해제 방법.

청구항 9.

제 8 항에 있어서,

상기 데이터의 상기 일부분은 비트이고,

상기 처리된 데이터의 양은 1 비트인 데이터 압축해제 방법.

청구항 10.

제 6 항에 있어서,

상기 데이터의 상기 압축해제된 일부분을 저장하는 단계를 더 포함하는 데이터 압축해제 방법.

청구항 11.

데이터를 압축하는 장치에 있어서,

산술 압축기(arithmetic compressor)를 포함하되,

상기 산술 압축기는,

확률표(probability table)와,

제 1 콘텍스트 모델과,

제 2 콘텍스트 모델과,

상기 산술 압축기에 접속되며, 상기 데이터를 수신하도록 배열된 시프트 레지스터와,

상기 산술 압축기에 접속된 카운터 – 상기 카운터는 상기 데이터의 일부분이 사전결정된 양보다 적을 때 상기 제 1 콘텍스트 모델을 이용하여 상기 데이터의 상기 일부분을 압축하도록 상기 산술 압축기에 신호를 출력하며, 대안으로는, 상기 제 2 콘텍스트 모델을 이용하여 상기 일부분을 압축하도록 상기 산술 압축기에 신호를 출력함 –

를 포함하는 데이터 압축 장치.

청구항 12.

제 11 항에 있어서,

상기 제 1 콘텍스트 모델은 상기 사전결정된 양보다 적은 데이터의 양에 대해 최적화되고,

상기 제 2 콘텍스트 모델은 상기 사전결정된 양보다 적지 않은 데이터의 양에 대해 최적화되는 데이터 압축 장치.

청구항 13.

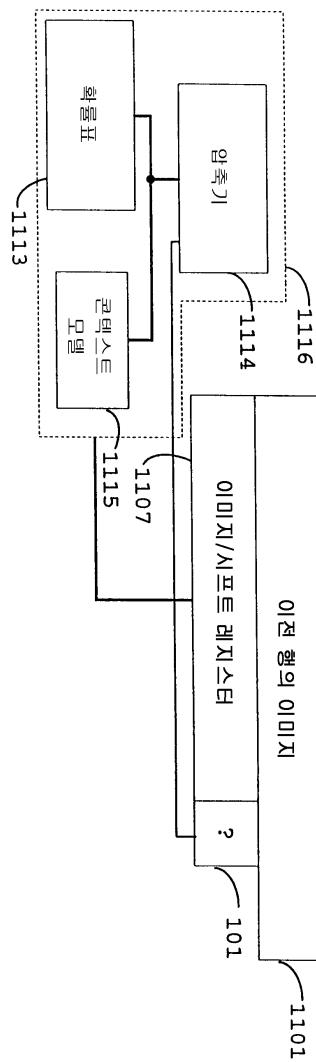
제 12 항에 있어서,

상기 제 1 콘텍스트 모델은 상기 시프트 레지스터의 내용을 상기 확률표의 제 1 부분으로 매핑하고,

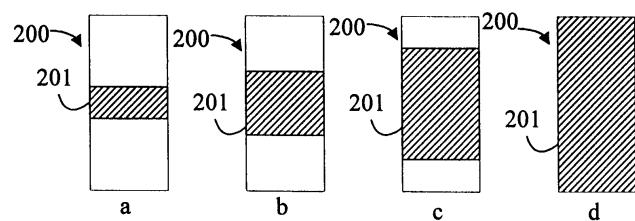
상기 제 2 콘텍스트 모델은 상기 시프트 레지스터의 내용을 상기 확률표의 제 2 부분으로 매핑하는 데이터 압축 장치.

도면

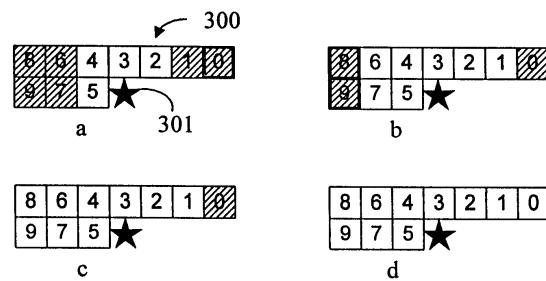
도면1



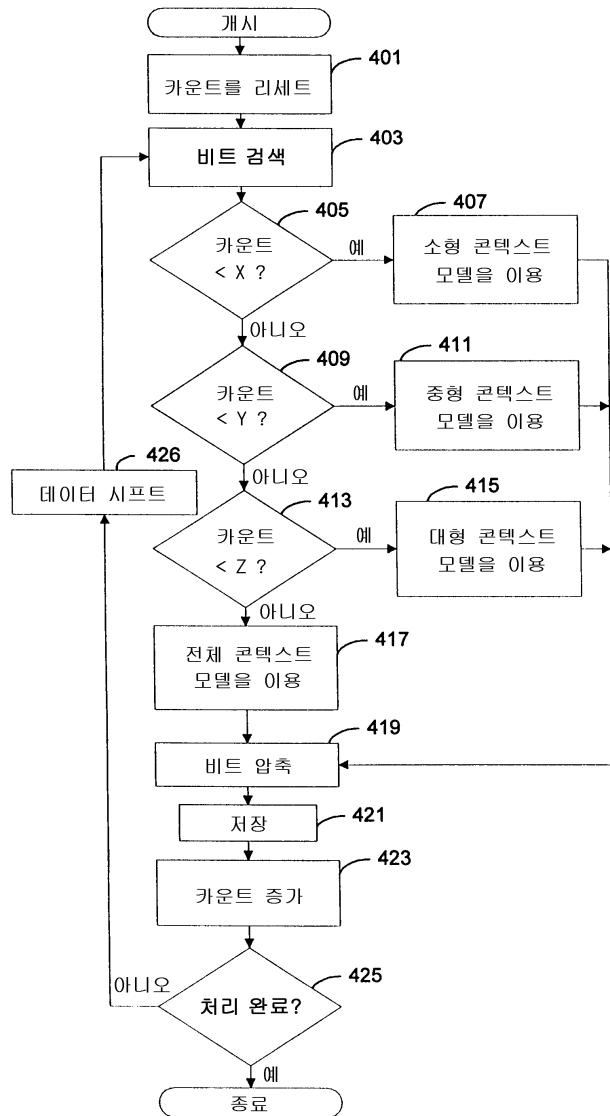
도면2



도면3



도면4



도면5

